UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/849,736	05/04/2001	G. Glenn Henry	CNTR:2021	8084

23669   7590   03/18/2004

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER
GOLE, AMOL V

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 03/18/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	**Applicant(s)**
Office Action Summary	09/849,736	HENRY ET AL.
	Examiner	**Art Unit**
	Amol V. Gole	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *5/4/01, 6/5/02, 11/18/02*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
   closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-42* is/are pending in the application.

   4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04 May 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a)☐ All   b)☐ Some * c)☐ None of:

   1.☐ Certified copies of the priority documents have been received.

   2.☐ Certified copies of the priority documents have been received in Application No. _____.

   3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
      application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
   Paper No(s)/Mail Date *2*.

4) ☐ Interview Summary (PTO-413)
   Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____ .

DETAILED ACTION

1.   Claims 1-42 have been examined.

Papers Submitted

2.   It is hereby acknowledged that the following papers have been received and

placed of record in the file:

   #2: IDS (6/5/02)

   #3: Change of Address (11/18/02)

Specification

3.   The lengthy specification has not been checked to the extent necessary to

determine the presence of all possible minor errors. Applicant's cooperation is

requested in correcting any errors of which applicant may become aware in the

specification.

4.   The summary is objected because:

   1) Change heading to "Brief Summary of Invention".

   2) The summary discloses information about many different related inventions.

   Please change it so that it discloses information only about that the invention with

   regard to which this application was filed.

5.   Please provide the serial numbers for the related applications in the "Related

Applications" section.

6.   The title of the invention is not descriptive. A new title is required that is clearly

indicative of the invention to which the claims are directed.

The following title is suggested: A SPECULATIVE BRANCH TARGET ADDRESS

CACHE THAT OUTPUTS THE TARGET ADDRESS OF BRANCH IN RESPONSE TO

A FETCH ADDRESS WHETHER OR NOT THE BRANCH IS PRESENT IN THE

INSTRUCTION CACHE LINE INDEXED BY THAT ADDRESS.

Oath/Declaration

7.   The oath or declaration is defective. A new oath or declaration in compliance

with 37 CFR 1.67(a) identifying this application by application number and filing date is

required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:

The Declaration refers to the application entitled "BRANCHING BASED ON

FETCH ADDRESS ONLY (BTAC) – NO ACTUAL BRANCH INSTRUCTION NEEDED

TO BRANCH" while the application submitted has a different title. Please resubmit the

Declaration with the proper title.

Claim Rejections - 35 USC § 102

8.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

> *A person shall be entitled to a patent unless –*
>
> *(b) the invention was patented or described in a printed publication in this or a*
>
> *foreign country or in public use or on sale in this country, more than one year*
>
> *prior to the date of application for patent in the United States.*

9.   Claims **1-7, 11, 13-15, 17-19, 21-30, and 34-42** are rejected under 35

U.S.C. 102(b) as being anticipated by Shiell et al. (US005850543A)

10.   **In regard to claim 1:**

11.   Shiell et al. disclose a branch target address cache (BTAC) (fig. 2, Branch Target

Buffer, BTB 56) for providing a speculative target address (col. 7, lines 40-42; the target

address is speculative because it is provided during the instruction fetch stage before it

is known whether the instruction is a branch or not) to address selection logic (fig. 2,

multiplexers 57, 58 and 52), the address selection logic selecting a fetch address for

addressing a line in an instruction cache (fig. 2 shows that the address selection logic

addresses the instruction cache 16; and col. 6, lines 16-23 indicate that the fetch

address addresses a stream of instruction data i.e. a line in the instruction cache), the

BTAC providing the speculative target address based on a presumption that a branch

instruction is present in the cache (fig. 2 shows that the BTB outputs a target address

D0-D127 indexed by the fetch address without knowing whether or not a branch is

present in the line of the instruction cache addressed by the fetch address because the

BTB and the instruction cache are accessed in parallel using the fetch address FA), the

BTAC comprising:

an array of storage elements, configured to cache target addresses of previously

executed branch instructions (fig. 2, BTB 56; col. 7, lines 40-42);

an input, coupled to said array, for receiving the fetch address, to index into said

array of storage elements to select one of said target addresses (fig. 2; col. 8, lines 12-

16); and

an output, coupled to said array, for providing said one of said target addresses

indexed by the fetch address to the address selection logic (fig. 2 shows the target

address D0-D127 are connected to an output to the address selection logic

[multiplexers 57, 58 and 52]);

wherein said output provides said one of said target addresses to the address

selection logic for selection as a subsequent fetch address whether or not a branch

instruction is present in the line of the instruction cache addressed by the fetch address

(fig. 2 shows that the BTB outputs a target address D0-D127 indexed by the fetch

address to the address selection logic (multiplexers 57, 58 and 52) without knowing

whether or not a branch is present in the line of the instruction cache addressed by the

fetch address because the BTB and the instruction cache are accessed in parallel using

the fetch address FA).

12.   In regard to claim 2:

13.   Shiell et al. disclose the branch target address cache of claim 1, wherein said

array of storage elements is further configured to store speculative branch information

(fig. 3, "T_n" and "HIS" fields; col. 8, lines 36-44, 57-67) associated with said previously

executed branch instructions.

14.   In regard to claim 3:

15.   Shiell et al. disclose the branch target address cache of claim 2, further

comprising:

a second output, coupled to said array, for providing a portion of said speculative

branch information to control logic for controlling the address selection logic in response

to said portion of said speculative branch information (although not shown, it is deemed

inherent to the BTB to have a second output which output a portion of the history

information indicating the outcome of the branch to control the address selection logic

multiplexers. This is because when the branch is indicated in the BTB as not taken, the

BTB target address should not be selected and the next sequential address should be

selected).

16.   In regard to claim 4:

17.   Shiell et al. disclose the branch target address cache of claim 2, wherein said speculative branch information comprises information predicting whether the branch instruction presumed present in the cache line will be taken (fig. 3, "HIS" field indicates that the branch will be taken if set to '111' or '110' [col. 8, lines 57-67]).

18.   In regard to claim 5:

19.   Shiell et al. disclose the branch target address cache of claim 4, wherein said information predicting whether the presumed branch instruction will be taken comprises a taken/not taken bit (fig. 9, when a conditional branch, the 2nd bit in the "HIS" field is a taken/not taken bit because when set to "1" it indicates taken, when set to "0" it indicates not taken).

20.   In regard to claim 6:

21.   Shiell et al. disclose the branch target address cache of claim 4, wherein said information predicting whether the presumed branch instruction will be taken comprises a plurality of bits (fig. 3, "HIS" field).

22.   In regard to claim 7:

23.   Shiell et al. disclose the branch target address cache of claim 6, wherein said plurality of bits is stored in a saturating up/down counter (col. 9, lines 5-18).

24.   In regard to claim 11:

25.   Shiell et al. disclose the branch target address cache of claim 2, wherein said

speculative branch information comprises information specifying a location within the

cache line of the branch instruction presumed present in the cache line (fig. 3, shows

the "T_n" field as an entry in the BTB. col. 8, lines 40-44 indicate that the T_n field holds

information specifying the location of a specific instruction within the cache line

associated with the logical address LA used to index into the BTB).

26.   In regard to claim 13:

27.   Shiell et al. disclose the branch target address cache of claim 2, wherein said

speculative branch information comprises an indication of a type of the branch

instruction presumed present in the cache line (fig. 2, "HIS" field, col. 8, lines 57-67).

28.   In regard to claim 14:

29.   Shiell et al. disclose the branch target address cache of claim 13, wherein said

indication of said type of the branch instruction indicates whether the branch instruction

is a call instruction ("HIS" = 011; col. 8, lines 57-67).

30.   In regard to claim 15:

31.   Shiell et al. disclose the branch target address cache of claim 13, wherein said

indication of said type of the branch instruction indicates whether the branch instruction

is a return instruction ("HIS" = 010; col. 8, lines 57-67).

32.   In regard to claim 17:

33.   Shiell et al. disclose the branch target address cache of claim 1, wherein each of

said storage elements is configured to cache a plurality of target addresses (col. 8, lines

22-27 indicates that there are 4 target addresses stored per storage element).

34.   In regard to claim 18:

35.   Shiell et al. disclose the branch target address cache of claim 1, wherein the

branch target address cache is external to the instruction cache (fig. 2).

36.   In regard to claim 19:

37.   Shiell et al. disclose a branch target address cache (BTAC) (fig. 2, Branch Target

Buffer, BTB 56) for caching characteristics of branch instructions only, the

characteristics including a branch target address (fig. 3, D_n) and prediction information

(fig. 3, HIS_n; col. 8, lines 57-67), the BTAC comprising:

an input, for receiving a fetch address that accesses an instruction cache that is

external to the BTAC (fig. 2; col. 8, lines 12-16);

an array of storage elements, coupled to said input and indexed by said fetch

address, for caching characteristics of branch instructions only (col. 7, lines 40-45; col.

8, lines 12-14) ; and

an output, coupled to said array, for providing a branch target address when said

input receives said fetch address (fig. 2 shows the target address D0-D127 are

connected to an output. Although not explicitly mentioned, the target address can be provided only when the input receives a fetch address to index into the BTB because otherwise there would be no way to know which target address to output);

wherein said branch target address is provided to said instruction cache as a subsequent fetch address (col. 7, lines 40-45 indicate that the target address is used to generate the fetch address and fig. 2 shows that this fetch address is provided to the instruction cache).

38.   In regard to claim 21:

39.   Shiell et al. disclose a pipelined microprocessor (fig. 1) having separate caches for instructions and branch target addresses (fig. 2, instruction cache 16$_i$ and BTB 56), the microprocessor comprising:

a first plurality of cache lines for storing instruction bytes (col. 8, lines 2-3 indicate that each address addresses a block of sixteen instructions in the instruction cache i.e. a cache line), said first plurality addressed by a fetch address on a fetch address bus (fig. 2); and

a second plurality of cache lines (col. 8, lines 22-24 indicates that there are 127 cache lines in the BTB), coupled to said fetch address bus (fig. 2), for storing branch target addresses (fig. 3 shows target addresses stored) that are addressed by said fetch address (fig. 2).

40.   In regard to claim 22:

41.   Shiell et al. discloses the microprocessor of claim 21, wherein said first and

second plurality of cache lines are physically distinct (fig. 2).

42.   In regard to claim 23:

43.   Shiell et al. disclose the microprocessor of claim 21, wherein said fetch address

is a virtual address (logical address col. 8, line 13).

44.   In regard to claim 24:

45.   Shiell et al. discloses the microprocessor of claim 23, wherein said first plurality

of cache lines are comprised in an instruction cache that has logic for translating said

virtual fetch address to a physical fetch address (fig. 2, µTLB 22; col. 7, lines 55-59),

wherein said second plurality of cache lines are comprised in a branch target address

cache (BTAC) that does not include logic for translating said virtual fetch address to a

physical fetch address (fig. 2 shows that the BTB 56 is addressed with the logical

address and there is not translating logic).

46.   In regard to claim 25:

47.   Shiell et al. disclose the microprocessor of claim 24, wherein said instruction

cache provides one of said first plurality of cache lines of instruction bytes selected

based on said physical fetch address, wherein said BTAC provides one of said target

addresses based on said virtual fetch address (col. 7, lines 59-62 and fig. 2 shows that

the logical fetch address is used to address the BTB to provide one of the target

addresses D0-D127).

48.   In regard to claim 26:

49.   Shiell et al. discloses the microprocessor of claim 21, wherein the

microprocessor speculatively branches to one of said target addresses addressed by

said fetch address even though one of said first plurality of cache lines of instruction

bytes addressed by said fetch address has been modified since said one of said target

addresses was cached in said second plurality of cache lines such that no branch

instructions are present in said addressed one of said first plurality of cache lines (fig. 2

shows that the logical fetch address addresses the BTB (2nd cache lines) to output a

target address in the fetch stage before decoding the instruction without knowledge of

whether the instruction at that fetch address in the instruction cache (1st cache lines) is

still a branch or not i.e. speculative execution [col. 8, lines 13-16] because the BTB and

the instruction cache are accessed in parallel using the fetch address FA).

50.   In regard to claim 27:

51.   Shiell et al. disclose the microprocessor of claim 21, wherein the microprocessor

is configured to speculatively branch to one of said branch target address addressed by

said fetch address in response to a hit of said fetch address within said second plurality

of cache lines whether or not a branch instruction is cached within one of said first plurality of cache lines of instruction bytes selected by said fetch address (fig. 2 shows that the logical fetch address addresses the BTB (2nd cache lines) to output a target address as the next fetch address during the fetch stage before decoding the instruction without knowledge of whether the instruction at that fetch address in the instruction cache (1st cache lines) is still a branch or not i.e. speculative execution [col. 8, lines 13-16] because the BTB and the instruction cache are accessed in parallel using the fetch address FA).

52.   In regard to claim 28:

53.   Shiell et al. disclose the microprocessor of claim 21, wherein a possibility exists of virtual aliasing of said fetch address between said first and second plurality of cache lines (Although not explicitly mentioned, this is deemed inherent to the microprocessor because the logical (virtual) address is used to address the BTB while the physical address is used to address the instruction cache).

54.   In regard to claim 29:

55.   Shiell et al. discloses the microprocessor of claim 21, wherein said first plurality of cache lines provides an instruction in response to said fetch address, wherein the microprocessor speculatively branches to one of said branch target addresses selected by said fetch address erroneously because said instruction is not a branch instruction.

56.   In regard to claim 30:

57.   Shiell et al. disclose the microprocessor of claim 21, further comprising:

an instruction buffer (fig. 2, element 60), coupled to said first plurality of cache

lines (instruction cache 16ᵢ), for buffering said instruction bytes received from said first

plurality of cache lines (col. 7, lines 65-67, col. 8, lines 1-4), wherein said instruction

buffer operates in conjunction with said second plurality of cache lines (BTB 56) to

accomplish substantially zero penalty speculative branches (fig 2 and col. 7, lines 40-

45, 53-67, col. 8, lines 1-4 disclose that the target address generated by the BTB is sent

to the instruction cache which responds with the speculative instruction block to be sent

to the instruction buffer. Therefore there is no stalling for the generation of the target of

the branch instruction resulting in zero-penalty speculative branches).

58.   In regard to claim 34:

59.   Shiell et al. disclose a method of speculatively branching in a pipelined

microprocessor (fig. 1), comprising:

caching a plurality of branch target addresses in a branch target address cache

(BTAC) (fig. 2, BTB 56; col. 7, lines 40-42);

accessing said BTAC with a fetch address of an instruction cache after said

caching (fig. 2 indicates that the fetch address FA is used to access the BTB 56; col. 8,

lines 13-16 indicate that the accessing is done when branch history is stored in the

BTB);

determining whether said fetch address hits in said BTAC in response to said

accessing (col. 8, lines 13-16); and

branching the microprocessor to one of said plurality of branch target addresses

selected by said fetch address if said fetch address hits in said BTAC whether or not a

branch instruction is cached in a line of said instruction cache indexed by said fetch

address (col. 7, lines 40-45 indicate that the target address is used to generate the fetch

address during the fetch stage before the decoding the instruction without knowledge of

whether the instruction at that fetch address in the instruction cache is still a branch or

not i.e. speculative execution [col. 8, lines 13-16] because the BTB and the instruction

cache are accessed in parallel using the fetch address FA).

60.   In regard to claim 35:

61.   Shiell et al. disclose the method of claim 34, further comprising:

storing a branch direction prediction associated with each of said plurality of

branch target addresses prior to said accessing said BTAC (fig. 3, "HIS_n" field; col. 8,

lines 57-67).

62.   In regard to claim 36:

63.   Shiell et al. disclose the method of claim 35, wherein said branching the

microprocessor to said one of said plurality of branch target addresses selected by said

fetch address is performed only if said associated branch direction prediction indicates

said branch instruction will be taken (Although not explicitly mentioned, the limitation is

deemed inherent to the correct functioning of the method because the purpose of the

prediction information when indicating that the speculative branch is taken is for

instructing the processor to branch to the target address of the branch and not the next

sequential address).

64.   In regard to claim 37:

65.   Shiell et al. disclose the method of claim 34, further comprising:

storing an indication said branching was performed if said branching is performed

(col. 7, lines 40-45, 65-67; col. 8, lines 1-4 indicate that if branching is performed by

using the target address as the fetch address, the instruction from the branch target

path is fetched and stored in the instruction buffer hence indicating that branching is

performed).

66.   In regard to claim 38:

67.   Shiell et al. disclose the method of claim 37, wherein said storing said indication

said branching was performed comprises storing said indication in an instruction buffer

(col. 7, lines 40-45, 65-67; col. 8, lines 1-4 indicate that if branching is performed by

using the target address as the fetch address, the instruction from the branch target

path is fetched and stored in the instruction buffer hence indicating that branching is

performed).

68.   In regard to claim 39:

69.   Shiell et al. disclose a method for speculatively branching in a pipelined

microprocessor (fig. 1), comprising:

providing a cached speculative branch target address without having decoded an

instruction for which said speculative branch target address is cached (col. 7, lines 40-

45);

providing a stored speculative branch direction without having decoded said

instruction for which said speculative branch direction is stored (col. 8, lines 57-67);

speculatively branching the microprocessor to said speculative branch target

address if said speculative branch direction indicates said instruction will be taken

(Although not explicitly mentioned, the limitation is deemed inherent to the correct

functioning of the method because the purpose of the prediction information when

indicating that the speculative branch is taken is for instructing the processor to branch

to the target address of the branch and not the next sequential address. This branching

is speculative because it occurs in the fetch stage, fig. 2 in parallel with the fetching in

the instruction cache).

70.   In regard to claim 40:

71.   Shiell et al. disclose a branch target address cache (fig. 2, BTB 56) for

speculatively predicting target addresses (col. 7, lines 40-45 indicates the BTB provides

target addresses in the fetch stage before decoding instruction hence making prediction

speculative) of branch instructions cached in an instruction cache (fig .2, 16_i),

comprising:

an input, for receiving a fetch address of the instruction cache (fig. 2; col. 8,

lines13-16);

an array of storage elements coupled to said input, each configured to cache a

target address of a branch instruction (col. 7, lines 40-42); and

an output, coupled to said array, for providing said target address cached in one

of said array of storage elements indexed by said fetch address (fig. 2 shows that the

target address D0-D127 indexed by the fetch address FA is output from the BTB array);

wherein said output provides said target address without said branch instruction

having been decoded by a microprocessor comprising said branch target address

cache (col. 7, lines 40-45 indicates the BTB provides target addresses in the fetch stage

before decoding instruction in parallel with the fetching in the instruction cache hence

making prediction speculative).

72.   In regard to claim 41:

73.   Shiell et al. disclose a pipelined microprocessor (fig. 1) for speculatively

branching, comprising:

an instruction cache (fig. 2, 16ᵢ), indexed by a fetch address provided on a fetch

address bus (FA bus output of multiplexer 52), said instruction cache providing a line of

instructions to instruction decode logic (col. 6, lines 16-23 indicate that the fetch address

addresses a stream of instruction data i.e. a line in the instruction cache to the

predecode stages 28 and 32 which further provide the instructions to decode stage 34,

fig. 1);

said instruction decode logic configured to decode said line of instructions

subsequent to said instruction cache providing said line of instructions (col. 6, lines 38-

41); and

a branch target address cache (fig. 2, BTB 56), coupled to said fetch address bus

(FA bus output of multiplexer 52), configured to receive said fetch address and in

response thereto to provide a speculative target address as a subsequent fetch address

on said fetch address bus (fig. 2 and col. 7, lines 40-45);

wherein the microprocessor is configured to speculatively branch to said

speculative target address prior to said instruction decode logic decoding said

instruction (col. 7, lines 40-45 indicates the BTB provides target addresses in the fetch

stage before decoding instruction in parallel with fetching in the instruction cache hence

speculatively branching).

74.   In regard to claim 42:

75.   Shiell et al. disclose the microprocessor of claim 41, wherein said instruction

decode logic decodes said line of instructions subsequent to the microprocessor

speculatively branching to said speculative target address (col. 7, lines 40-45 indicates

the BTB provides target addresses in the fetch stage before decoding instruction in

parallel with fetching from the instruction cache hence speculatively branching. Further

col. 7, lines 65-67, col. 8, lines 1-4, and fig. 1 indicate that the instruction cache provides

the line of instructions (block of instructions) to the predecode stages and decode

stages for decoding) and said instruction decode logic determines that no branch

instructions are present in said line of instructions (Although not explicitly mentioned, it

is deemed inherent to the decoding method that the decoder will determine that there

are no branch instructions present in the said line of instructions when the instructions

presented to the decoder do not contain any branch instructions because col. 6, lines

38-42 indicate that the decode stage is responsible for decoding the instructions i.e.

determining the type of instructions).

Claim Rejections - 35 USC § 103

76.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed
>
> or described as set forth in section 102 of this title, if the differences between the
>
> subject matter sought to be patented and the prior art are such that the subject
>
> matter as a whole would have been obvious at the time the invention was made
>
> to a person having ordinary skill in the art to which said subject matter pertains.

Patentability shall not be negatived by the manner in which the invention was made.

77.   Claims **8-10** are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et al. (US005850543A) in view of Bae et al. (US006044459A).

78.   In regard to claim 8:

79.   Although Shiell et al. verifies whether the target address was correct by comparing it with the actual next instruction address determined by the execution unit (Shiell: col. 2, lines 48-51) they do not disclose the branch target address cache of claim 3, wherein said portion of said speculative branch information comprises an indication of whether said one of said target addresses is a valid target address.

80.   Bae et al. disclose a BTB entry format with a valid bit for indicating whether the target address is a valid target address (col. 5, lines 27-34). The valid bit is checked before providing the target address as the fetch address so that an invalid target is not incorrectly provided.

81.   It would have been obvious to one of ordinary skill in the art at the time of the invention to recognize to add a valid bit in the BTB of Shiell et al. indicating the validity of the target address so that in invalid target is not provided.

82.   One would have been motivated to do so because by including the valid bit incorrect targets are not fetched and subsequent flushing of the pipeline is not required leading to improved performance.

83.   In regard to claim 9:

84.   The combination of Shiell et al. in view of Bae et al. disclose the branch target address cache of claim 8, wherein said indication is populated to indicate said one of said target addresses is a valid target address in response to execution of the presumed branch instruction, wherein said one of said target addresses is resolved (This limitation is deemed inherent because the valid bit can be set to indicate the validity of the predicted target address only after the branch instruction is executed and the target is calculated).

85.   In regard to claim 10:

86.   The combination of Shiell et al. in view of Bae et al. disclose the branch target address cache of claim 8, wherein said indication is populated to indicate said one of said target addresses is not a valid target address in response to detecting said one of said target addresses is erroneous subsequent to said providing said one of said target addresses on said output (This limitation is deemed inherent because the valid bit can be set to indicate that the predicted target address is invalid only after comparing the predicted target address in the BTB with the target address calculated on execution i.e. detecting that the target address is erroneous).

87.   Claims **12 and 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et al. (US005850543A).

88.   In regard to claim 12:

89.   Shiell et al. disclose the branch target address cache of claim 2, wherein said speculative branch information comprises data corresponding to the branch instruction presumed present in the cache line (fig. 3, "T_n" field indicates the position of the branch instruction in the cache line).

90.   Shiell et al. does not expressly show that the information comprises "the length" of the branch instruction presumed present in the cache line.

91.   However this difference is only found in the nonfunctional descriptive material and are not involved in the functioning of the BTB. Thus this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, see In re Gulack, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); In re Lowry, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994).

92.   Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to store any kind of data in the branch prediction information because such data does not alter how the BTB functions and because the subjective interpretation of the data does not patentably distinguish the claimed invention.

93.   In regard to claim 16:

94.   Shiell et al. disclose the branch target address cache of claim 2, wherein said speculative branch information comprises an indication of the branch instruction presumed present in the cache line (fig. 3, "T_n" field indicates the position of the branch instruction in the cache line).

95.   Shiell et al. does not expressly show that the information comprises an indication

of whether the branch instruction presumed present in the cache line "spans more than

one line in the instruction cache".

96.   However this difference is only found in the nonfunctional descriptive material

and are not involved in the functioning of the BTB. Thus this descriptive material will not

distinguish the claimed invention from the prior art in terms of patentability, see In re

Gulack, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); In re Lowry, 32

F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994).

97.   Therefore, it would have been obvious to one of ordinary skill in the art at the

time of the invention to store any kind of indication in the branch prediction information

because such data does not alter how the BTB functions and because the subjective

interpretation of the data does not patentably distinguish the claimed invention.

98.   Claim **20** is rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et

al. (US005850543A)

99.   In regard to claim 20:

100.   Shiell et al. disclose a pipelined microprocessor (fig. 1) having a branch target

address cache (fig. 2, Branch Target Buffer, BTB 56), the microprocessor comprising:

        first cache lines, within the branch target address cache, for caching branch

target addresses (col. 8, lines 22-24 indicates that there are 127 cache lines in the BTB;

fig. 3 shows target addresses cached);

second cache lines, within an instruction cache, for caching instructions (col. 8, lines 2-3 indicate that each address addresses a block of sixteen instructions i.e. a cache line);

wherein said first cache lines and said second cache lines are coupled to a fetch address bus that provides a fetch address for indexing into both of said first and second cache lines (fig. 2).

101. However, while Shiell et al. mention that there are 127 cache lines in the BTB, they do not mention that the number of said first cache lines is less than the number of said second cache lines.

102. "Official Notice" is taken that it is well known and expected to have a less number of BTB cache lines than the instruction cache because it would lead to saving in space because every cache line may not contain a branch.

103. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to have a BTB with a less number of cache lines than the instruction cache as one would recognize that by having a smaller BTB one would save expensive die area.

104. Claims **31-33** rejected under 35 U.S.C. 103(a) as being unpatentable over Shiell et al. (US005850543A) and Dietz et al. (US005634103).

105. In regard to claim 31:

106. Shiell et al. teach a pipelined microprocessor comprising:

an instruction cache (fig. 2, 16$_i$) that is indexed by a fetch address, said

instruction cache for caching instructions (col. 5, lines 37-38), and for providing said

instructions to an instruction buffer (fig 2, 60);

a branch target address cache (fig. 2, BTB 56), coupled to said instruction buffer

and indexed by said fetch address (fig. 2), for caching branch target addresses (col. 7,

lines 40-42);

107.   Shiell et al. do not disclose that the instruction buffer comprises a plurality of hit

indicators that are associated with said instructions, said indicators specifying whether

the microprocessor has speculatively branched to one of said branch target addresses.

108.   Dietz et al. teaches processor in which each instruction has a corresponding

speculative bit indicating whether the instruction is within the speculative execution

path. If the speculative path is resolved as incorrect, the instruction tagged as

speculative are flushed (col. 4, lines 60-67).

109.   One of ordinary skill in the art would have recognized that by adding hit indicators

to the instructions in the instruction buffer indicating whether they are on the

speculatively branched path or not, the process of flushing the speculative instructions

on a misprediction simply involves flushing the instructions tagged as speculative.

Moreover Shiell et al. mentions that instructions along the speculative path must be

flushed on determining that the prediction was not correct.

110.   Hence it would have been obvious to one of ordinary skill in the art at the time of

the invention to modify the processor of Shiell et al. by adding a plurality of hit indicators

to each instruction of the instruction buffer indicating whether the processor

speculatively branched.

111.   One would have been motivated to do so because it would simplify identifying the

instructions in the pipeline that are speculative and those that are not speculative while

flushing the pipeline in the case of a mis-prediction leading to simpler logic design.

112.   In regard to claim 32:

113.   The combination of Shiell et al. in view of Dietz et al. teaches the microprocessor

of claim 31, wherein said instruction buffer includes one of said plurality of hit indicators

associated with each byte of said each of said instructions stored in said instruction

. buffer (Dietz et al. teaches the use of the hit indicator for **each** instruction [col. 4, lines

60-65]).

114.   In regard to claim 33:

115.   The combination of Shiell et al. in view of Dietz et al. teaches the microprocessor

of claim 31, wherein said instruction cache and said branch target address cache are

accessed substantially in parallel (Shiell et al. teaches that the fetch address is

presented to various functions to control the fetching of the next instruction to be

decoded [col. 7, lines 53-55]. fig. 2, col. 7, lines 59-62, and col. 8, lines 13-16 indicate

that both the instruction cache and the BTB are accessed substantially in parallel).

Conclusion

116.   The prior art made of record and not relied upon is considered pertinent to

applicant's disclosure. Applicant is reminded that in amending in reply to a rejection of

claims in an application or patent under reexamination, the applicant or patent owner

must clearly point out the patentable novelty, which he or she thinks the claims present

in view of the state of the art disclosed by the references cited or the objections made.

The applicant or patent owner must also show how the amendments avoid such

references or objections. See 37 CFR § 1.111.

    a.   Emma et al. (US005353421A) teaches a speculative BTB.

    b.   Hoyt et al. (US005812839A) teaches a hybrid prediction scheme where

the BTB receives speculative branch prediction information from a second

prediction mechanism using the fetch address. It also shows a return stack

buffer.

    c.   Black et al. (US005530825A) teaches a speculative BTB.

    d.   Bray and Flynn ("Strategies for Branch Target Buffers," TR No. CSL-TR-

91-480, Stanford University, June 1991) teaches design criteria for selecting

various BTB configurations.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Amol V. Gole whose telephone number is 703-305-

8888. The examiner can normally be reached on 9:00-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Eddie Chan can be reached on 703-305-9712. The fax phone number for

the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

AVG
amol.gole@uspto.gov

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100